Bang for your Buck: Performance Impact Across Choices in Learning Architectures for Crisis Informatics

Shivam Sharma
New Jersey Institute of Technology
ss4354@njit.edu

Cody Buntain
New Jersey Institute of Technology
cbuntain@njit.edu

ABSTRACT

Over the years, with the increase in social media engagement, there has been an in increase in various pipelines to analyze, classify and prioritize crisis-related data on various social media platforms. These pipelines utilize various data augmentation methods to counter imbalanced crisis data, sophisticated and off-the-shelf models for training. However, there is a lack of comprehensive study which compares these methods for the various sections of a pipeline. In this study, we split a general crisis-related pipeline into 3 major sections, namely, data augmentation, model selection, and training methodology. We compare various methods for each of these sections and then present a comprehensive evaluation of which section to prioritize based on the results from various pipelines. We compare our results against two separate tasks, information classification and priority scoring for crisis-related tweets. Our results suggest that data augmentation, in general,improves the performance. However, sophisticated, state-of-the-art language models like DeBERTa only show performance gain in information classification tasks, and models like RoBERTa tend to show a consistent performance increase over our presented baseline consisting of BERT. We also show that, though training two separate task-specific BERT models does show better performance than one BERT model with multi-task learning methodology over an imbalanced dataset, multi-task learning does improve performance for more sophisticated model like DeBERTa with a much more balanced dataset after augmentation.

Keywords

Incident Streams, TREC, TRECIS, crisis informatics

INTRODUCTION

Since its inception in 2018, the Incident Streams track at the annual Text Retrieval Conference(TREC-IS) has received various submissions for the information classification and priority scoring tasks. These submissions are generated from pipelines that utilize state-of-the-art language models for result generation, various augmentation techniques to counter imbalance in the data, and various other methods which might help elevate the efficiency of their pipelines. However, there is a need for a comparison between different methods that can be used within different sections of a pipeline. This work aims to fill this gap by comparing different methods in three major sections of a pipeline, namely, data augmentation, model selection, and model training. This work spans TREC-IS 2021-A and 2021-B, with the results being compared on the 2021-A test dataset.

In crisis informatics, data sparsity, model selection and learning methodology are some crucial factors to take into consideration during pipeline development, and while numerous methods exist in each sections, little domain-specific guidance exists for choosing or prioritizing these approaches. When considering these different sections of a crisis informatics pipeline, there is a lack of guidance regarding prioritization of these sections for optimization. For example, using an off-the-shelf, state-of-the-art pre-trained language model may showcase some performance gain, however, an older model with better class-balanced dataset might show similar or better performance, which highlights the need of prioritization of the various sections of a pipeline. This paper provides this much needed domain-specific guidance by evaluating performance improvements across a series of data augmentations, model improvements, and learning designs.

In particular, this paper aims to answer the following research questions:

RQ1: Across several augmentation techniques, what is the expected performance increase, on average and for each method, when applied to crisis-informatics data?

RQ2: By how much might one expect performance to increase by using increasingly sophisticated off-the-shelf, pre-trained models?

RQ3: How much performance increase is observed upon using a single multi-task learning pipeline, as compared to different task-specific pipelines?

RQ4: Across data augmentation, language model selection, and training methodology, which choices should be prioritized for optimization for maximum performance gain?

To answer these question, this paper presents a systematic comparison of test results showcased by different pipelines. We outline a core pipeline architecture and make changes in the sections we aim to compare, leaving the rest constant. We define a single baseline model, which will aid us in comparison across different research questions and also help identify the section of pipeline which gives us the most improvements.

Our results show that, amongst the three different augmentation strategies used(EDA, AugLy, Synonym-Replacement), even though there is meaning in using augmentation, there is no clear winner. Our results also showcase that using off-the-shelf pre-trained models does improve the performance by a degree in tasks, however, for models like BERT there is little improvement when switching the learning techniques from single, task-specific learning to multi-task learning. Though using sophisticated models like DeBERTa do showcase improved performance when switching from single, task-specific learning to multi-task learning.

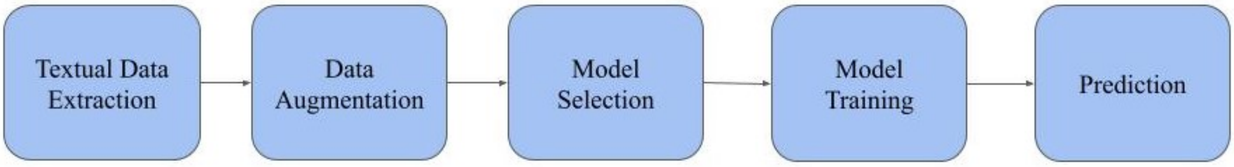

Figure 1. Core Architecture. We split a general pipeline into three major sections, namely, data augmentation, model selection and model training. To answer the research questions defined in this work, we change these sections to create new pipelines.

Primary contribution of this work will be of interest to those studying crisis-informatics and methods to improve the performance of their pipelines by giving an insight into which section to prioritize for optimization. Crisis Informatics researchers in particular can benefit from the comparison of different learning methods and different language models.

RESEARCH QUESTION

In this section, we dive deeper into the various research questions and outline the methods used to answer these questions.

Augmentation Comparison

Like many other real-world data, the crisis tweets collected by TREC-IS are imbalanced in the distribution across different information as well as priority labels. This is expected as there would be a higher number of people posting a tweet showing support or giving condolences to the victims of some crisis events as opposed to the number of people requesting aid for the same crisis. To counter this imbalance various augmentation strategies can be applied to the dataset.

Related Work

It is well known that ingestion of addition data or noise to the original training data of a neural network can lead to significant improvements in performance. Bishop 1995 show that training with noised example is reducible to a form of regularization (Tikhonov regularization). However, recent years have seen a boost in data augmentation techniques in the field of Natural Language Processing (NLP). Various works by Kobayashi 2018, Wei and Zou 2019, and X. Zhang et al. 2015 discuss the significant performance improvement in various tasks upon using text augmentation like word replacement with semantically similar word, random insertion, random deletion, etc. However there is little domain specific comparison for various augmentation methods. In this work, we aim to bridge this gap by comparing three different augmentation methods (EDA, Synonym-Replacement, and AugLy) for on domain specific dataset.

Experiment Description

In this experiment, we aim to answer the following research question:

RQ: Across several augmentation techniques and libraries, how much performance increase is observed, on average as well as for each technique or library, when applied to crisis informatics data?

We answer the above mentioned research question by observing the performance improvement across the following augmentation strategies for the textual data present in the tweets:

1. **Synonym-Replacement:** In this method, we list all the verbs in the given tweet text and replace them with their synonyms, thus "generating" new tweet text. If a text has more than one verb in it, then we replace one verb with its respective synonyms at a time, keeping the rest of the verbs the same. We replace only the verb portion of the tweet since we assume the verbs to contain the majority of information in any given tweet.
2. **Easy Data Augmentation:** Easy Data Augmentation, or EDA, is the work presented by Wei and Zou Wei and Zou 2019. This method includes four different text processing methods, namely, Synonym-Replacement, random insertion, random swap, and random deletion. This method is a more advanced version of our Synonym-Replacement method, with Synonym-Replacement in any random word instead of just the verb.
3. **AugLy:** AugLy is a data augmentation library recently developed by Facebook Research. It contains over 100 different augmentation methods across multi-modalities like image, text, video, and audio. For textual data augmentation, AugLy introduces 11 different augmentation functions, of which we have used three for this experiment, namely, word splitting, similar character replacement, and "typos" simulation. We use these three out of the original eleven since these are the only text-based augmentation methods, with the other methods including punctuation after every letter, changing the font of text, flipping random words upside down. These methods have a higher chance of changing the context of the text.

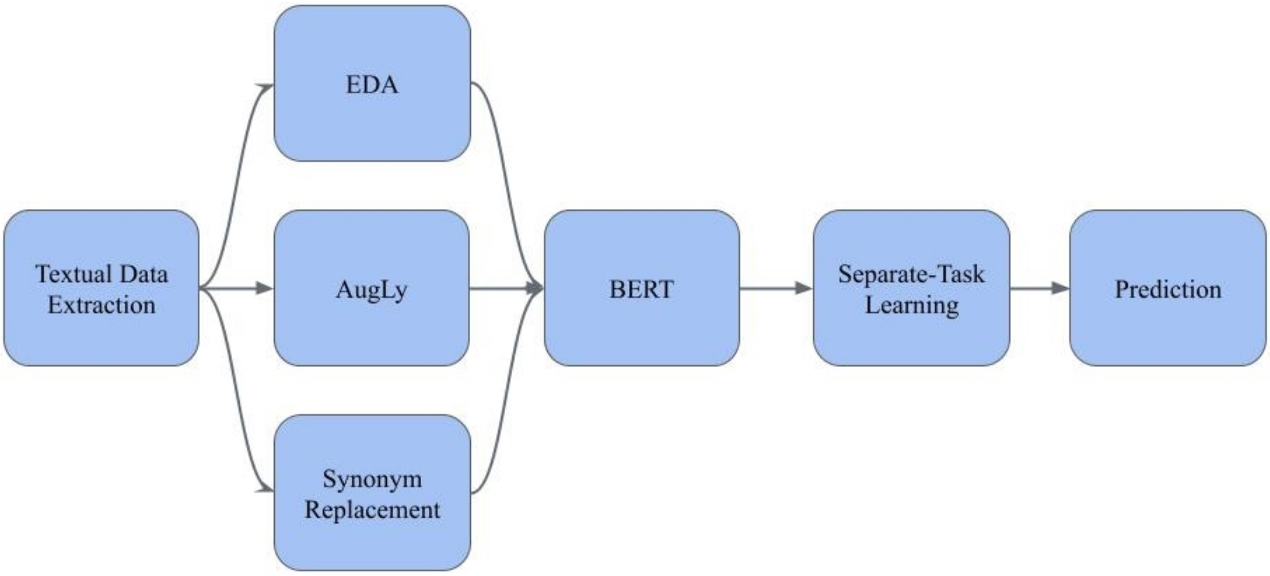

Figure 2. Augmentation Pipelines. We use three different augmentation methods, namely, EDA, AugLy, and Synonym-Augmentation and compare the evaluation scores against our baseline which uses non-augmented, imbalanced data.

Figure 2 presents the three different pipelines used for this experiment. For all of these three augmentation strategies, we define a multiplication factor, which is the number of times a tweet text gets augmented. We also augment only the "actionable" classes, which means, we augment only those tweets which belong to classes which have high average priority. We augment only these "actionable" classes due to the fact that these classes have a higher average priority score as compared to the other classes, which makes tweets belonging to these information classes more important to response officers. Thus, including additional data for these classes would improve our pipeline's ability to identify tweets with higher priority, which is the one of its main objectives. We use BERT as the core language model for all three use cases and separate-task learning as the learning methodology, which means we train separate models for information classification task and priority scoring task.

Model Selection

Model selection is another important aspect to consider when formulating a pipeline. There are various pre-trained language models available which are trained on a high amount of data, like BERT, RoBERTa, XLM, DeBERTa, etc. The state-of-the-art limits are frequently pushed with better models, which are either trained on a greater amount of data or use a denser or more refined base architecture. These options raise the question of whether constantly updating our crisis informatics pipelines with these new, off-the-shelf pre-trained language model guarantee performance improvement.

Related Work

It has been established that deep learning methods show higher performance gain over classical machine learning methods for various disaster related social media tasks. Nguyen et al. 2016 and Caragea et al. 2016 showcase the efficiency of neural models like CNNs over classical machine learning models like SVMs. Work done by Neppalli et al. 2018 also showcase a similar study in which they compare the performance of classical machine learning models like Naive Bayes with neural models like CNNs and RNNs.

For the past few years, attention based neural models called transformers have shown state-of-the-art results in various NLP tasks. Transformer models like BERT (Devlin et al. 2018), RoBERTa (Y. Liu et al. 2019) and newer models like DeBERTa (He et al. 2020) have showcased state-of-the-art results in various NLP tasks. Works like Chowdhury et al. 2020 and J. Liu et al. 2021 discuss the performance gain showcased by various transformer models on different crisis classification tasks. In this work, we aim to build on these works by comparing different transformer models in controlled environment trained over non-augmented, imbalanced crisis data. Through this we aim to analyze the efficiency of upgrading crisis-pipelines with off-the-shelf, state-of-the-art language models.

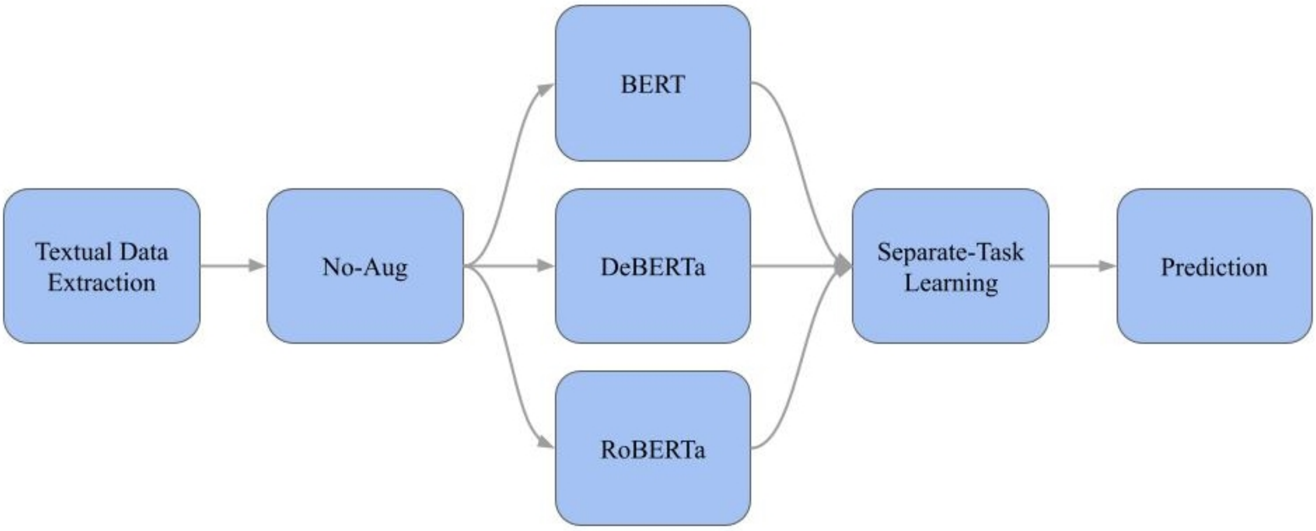

Figure 3. Model Selection Pipelines. We compare three different language models, namely, BERT, DeBERTa, and RoBERTa, with the BERT model being the baseline.

Experiment Description

Through this experiment, we aim to answer the following research question:

RQ: By how much might one expect performance to increase by using increasingly sophisticated off-the-shelf, pre-trained models?

We answer this question by comparing three different language models, namely, BERT, RoBERTa, and DeBERTa, as presented in Figure 3, to check whether using state-of-the-art models like DeBERTa showcases improved results as compared to the base models like BERT and RoBERTa. In this experiment, we change the pre-trained language model in our base pipeline. As evident from the Figure 3, we use no augmentation on the data, and we use separate-task learning as the learning methodology, which means we train separate models for information classification task and priority scoring task.

Learning Methodology

C. Wang et al. 2021 in their work use multi-task learning, using a combined loss from priority scoring task model and information classification model. This was in turn motivated by Y. Zhang and Yang 2017, whose work shows evidence that parameter sharing between multiple tasks is likely to enable one task to share its learned knowledge with another. Wang, in their work, defines a parameter lambda, λ, as a loss parameter which they use as a weight for adjusting the loss of priority scoring task and information labels classification to calculate the final loss. The following equation describes the calculation for the final loss function, where L_{it} is the loss for information type classification task and L_{pri} is the loss for priority scoring task.

$$L_{total} = \lambda L_{it} + (1 - \lambda) L_{pri}$$

Related Work

Multi-task learning has achieved significant results in various NLP tasks as discussed in a survey by Y. Zhang and Yang 2017. Various works like Sun et al. 2019 and Zheng, F. Wang, et al. 2017 present the performance gain showcased by various LSTM models while using multi-task learning. Similar to these, Zheng, Hao, et al. 2017 also present the performance gain showcased by BiLSTM when using multi-task learning for entity recognition and relation extraction tasks. Experiments done by Xue et al. 2019 showcase the performance improvement by BERT transformer model when using multi-task learning on medical domain. Similarly C. Wang et al. 2021 presents comparison between BERT, DistilBERT (Sanh et al. 2019), ALBERTA (Lan et al. 2019), and ELECTRA (Clark et al. 2020) for crisis-related data. However, a domain-specific evaluation with non-augmented, imbalanced crisis data is required. Through this work, we aim to bridge this gap by comparing the model performance showcased by BERT model on a non-augmented, imbalanced crisis data.

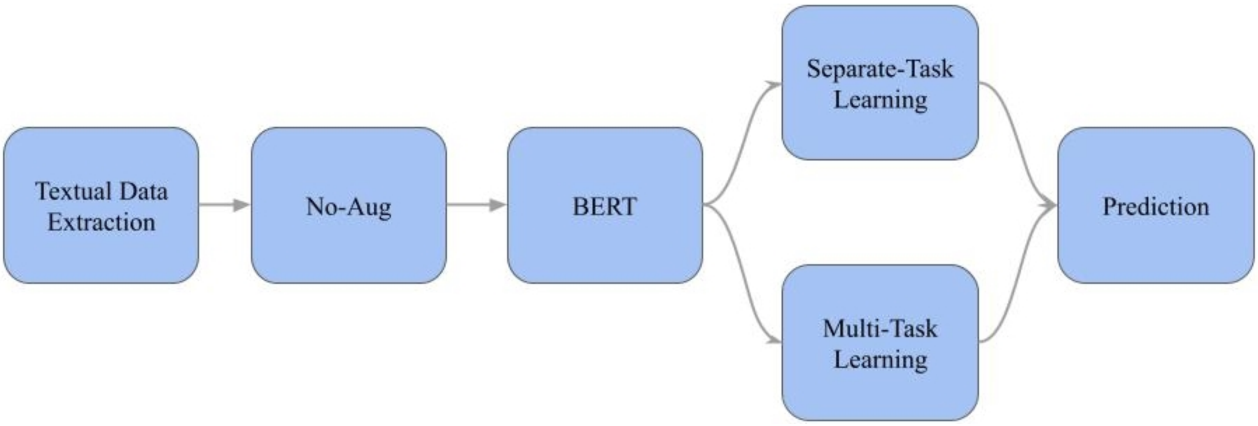

Figure 4. Learning Methods Pipelines. We compare two different learning methods, namely, separate-task learning and multi-task learning, with the separate-task learning being our baseline.

Experiment Description

Through this experiment we aim to answer the following research question:

RQ: How much performance increase is observed upon using single multi-task learning pipeline, as compared to different task-specific pipelines?

To answer this, we compare two different pipelines, one with multi-task learning, and one with two task-specific models, one for each, information type classification task and priority scoring task, as presented in Figure 4. We can get a model for priority scoring task only and information classification task only, if we set λ as 0 and 1, respectively. We use the λ parameter to train two different models, one for each task, and combine their results to form the final result. For multi-task learning, we average the losses for the two tasks, thus we use λ as 0.5. Our initial experiments with different values of λ showed no major improvement in the two tasks. As evident from Figure 4, we use no augmentation on the data and we use BERT as the core language model.

Section Prioritization

Over the past few years, with improvements in various NLP techniques, there has been parallel improvement in crisis-related tasks. However there is a need for domain-specific prioritization of these techniques for better development of crisis-related pipelines. Through this work, we aim to bridge this gap by comparing the various pipelines and recommending section prioritization based on the comparative results of these pipelines.

In this experiment, we aim to answer the following research question:

RQ: Across data augmentation, language model selection, and training methodology, which choices should be prioritized for optimization for maximum performance gain?

To answer this question, we compare the results of the following pipelines. Please note the names in the brackets represent the pipeline names as used in the tables. The initial “S" or “M" represent separate-task learning or multi-task learning, respectively. “De" represents DeBERTa language model and “Ro" represents RoBERTa language model. “EDA" at the end represents the use of EDA augmentation and the absence of it represents no augmentation.

- *DeBERTa + Separate + EDA [S-DeEDA]*: This pipeline uses EDA for augmentation, DeBERTa as its language model, and separate-task learning as the learning methodology.
- *DeBERTa + Multi [M-De]*: This pipeline does not use any augmentation method, takes DeBERTa as its language model, and uses multi-task learning as the learning methodology.
- *DeBERTa + Multi + EDA [M-DeEDA]*: This pipeline uses EDA as the augmentation method, DeBERTa as its language model, and multi-task learning as the learning methodology.
- *RoBERTa + Separate + EDA [S-RoEDA]*: This pipeline uses EDA as the augmentation method, RoBERTa as its language model, and separate-task learning as the learning methodology.
- *RoBERTa + Multi + EDA [M-RoEDA]*: This pipeline uses EDA as the augmentation method, RoBERTa as its language model, and multi-task learning as the learning methodology.
- *Best of RQ1*: These are the best results showcased in experiment for augmentation method comparison, for each individual evaluation score.
- *Best of RQ2*: These are the best results showcased in experiment for model selection,for each individual evaluation score.
- *Best of RQ3*: These are the best results showcased in experiment for learning methodology, for each individual evaluation score.
- *Best Run Results*: These are the best overall results for each individual evaluation score as showcased on the leadership board of the Incident Stream 2021 Track. [1].

METHODOLOGY

This section outlines the architectures of the basic architecture of the core pipeline, modifying different sections of which would generate the pipelines which we use to answer the research question. We also discuss in brief the dataset used for training and testing, and also the evaluations metrics used.

[1] Link to Incident Stream 2021 github repository: https://github.com/trecis/trecis.github.io

Task Description

The experiments presented in this work are evaluated on the basis of the following two tasks:

1. All High-Level Information Type Classification: Systems participating in this task are given tweet streams from a collection of crisis events and should classify each tweet as having one or more of the high-level information types (Buntain et al. 2020). As this is a multi-label classification task, the system should be able to assign as many categories to any given tweet as appropriate.
2. Priority Scoring: Systems participating in this task are given a tweet streams from a collection of crisis events and are expected to produce a relative score based on how important the information contained in the tweet is for a response officer. The score ranges between 0 and 1 with 0 being low importance/priority and 1 representing high importance/priority.

Data Description

Since its inception in 2018, TREC-IS has accumulated a total labeled corpus of more than 60k labeled crisis tweets made during 75 different crisis events. These labeled tweets can be distributed into 5 different subsets, based on their editions, namely, 2018, 2019-A, 2019-B, 2020-A, and 2020-B. Tables 1 and 2 shows the per-topic split for the training data. For our experiments in this work, we use the whole dataset for training and we use the evaluation notebook provided for 2021-A submissions to compare the results of the various pipelines. The evaluation notebooks are compared across more than 20k tweets.

Table 1. Per-Topic Training Data Distribution for Actionable Classes

Topics	Label Count
CallToAction-MovePeople	761
Report-EmergingThreats	8195
Report-NewSubEvent	3569
Report-ServiceAvailable	2800
Request-GoodsServices	241
Request-SearchAndRescue	347

Table 2. Per-Topic Training Data Distribution for Non-Actionable Classes

Topics	Label Count
CallToAction-Donations	1173
CallToAction-Volunteer	286
Other-Advice	3711
Other-ContextualInformation	6283
Other-Discussion	6249
Other-Irrelevant	326665
Other-Sentiment	12590
Report-CleanUp	508
Report-Factoid	11847
Report-FirstPartyObservation	5663
Report-Hashtags	17364
Report-Location	28143
Report-MultimediaShare	27166
Report-News	22063
Report-Official	3692
Report-OriginalEvent	4569
Report-ThirdPartyObservation	19326
Report-Weather	7927
Request-InformationWanted	482

Evaluation Metrics

TREC-IS releases an evaluation notebook for every edition. These notebooks include, an nDCG metric for ranking content by priority, F1 score divided into two sets, one restricted to "actionable" information types, and the other containing all possible labels, and R score for priority score comparison which, similar to the F1 score, is also divided into two sets, one for "actionable" information type and one for all possible labels. This "actionable" set is restricted to the top six information types with the highest average priority score, making them the most "important" types to classify correctly in a qualitative sense. These evaluations are made on a subset of the test data, since the test data is annotated by humans. Thus the results we discuss are based off of a sample of the test case and might change when evaluated across a bigger sample.

We also compare F1 score split for each information class and F1 score split for each priority class. The priority scores can be split into four different sections:

1. **Critical:** Priority scores between 0.75 and 1.
2. **High:** Priority scores between 0.5 and 0.75.
3. **Medium:** Priority scores between 0.25 and 0.5.
4. **Low:** Priority scores less than 0.25.

Model Description

We keep a consistent model architecture across all the experiments described in this work. This works aim to compare the results between three different sections of a neural pipeline, namely, augmentation, pre-trained model, and training method. We change the specific sections of the basic neural pipeline architecture and keep everything else the same for a clear and consistent comparison.

Baseline Architecture

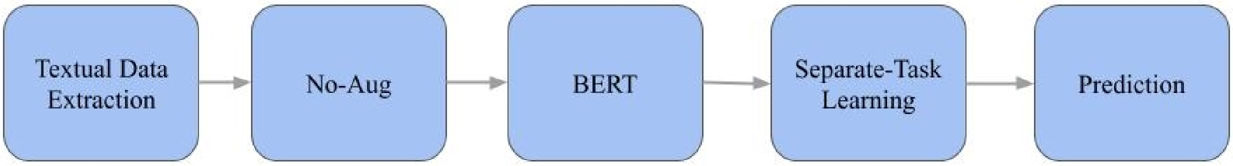

Figure 5. Baseline Architecture. For our baseline pipeline, we train the non-augmented, imbalanced data on BERT language model with separate-task learning as the learning methodology.

Our model architecture is heavily inspired by the model architecture used by the model described by C. Wang et al. 2021. Figure 1 describes our core pipeline architecture, which can be split into five major sections, namely, textual data extraction, data augmentation, model selection, model training, and prediction. Of these five, in this work, we change the data augmentation, the model selection, and the model training sections for their respective experiments.

Figure 5 describes the architecture for the baseline used to compare against the various experimental pipelines to evaluate the performance gain. The pipeline uses the original textual data without any augmentation, BERT as the language model and single-task learning, which means, there are two models trained, one for priority scoring task and the other for information classification task.

RESULTS

This section presents the results for the various experiments outlined above. We compare the various pipelines based on the evaluation scores as well as across individual information type scores. We also compare the net percent improvement from our baseline.

Augmentation Comparison

Table 3 shows the results from the various augmentation methods. As evident from the table, EDA augmentation shows improvement in information classification scores but shows a loss in performance in majority of the priority scores, except F1 score for actionable classes. Synonym-Replacement and AugLy showcase loss in performance in comparison to the baseline, which uses no augmentation method.

Table 3. Evaluation Score Comparison Between Augmentation Pipelines. Results show that EDA augmentation does show improvement in information classification task, however, baseline outperforms the augmentation pipelines in priority scoring task.

Augmentation Statergies	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline	**0.5043**	0.1958	0.2872	**0.8873**	0.2426	**0.2648**	**0.1978**	**0.229**
EDA	0.5037	**0.2728**	**0.312**	0.8864	**0.2586**	0.2469	0.1658	0.2276
Syn-Aug	0.4946	0.2193	0.2901	0.8865	0.2512	0.2464	0.1047	0.2092
AugLy	0.4917	0.2032	0.2979	0.8868	0.2289	0.2265	0.1903	0.2133

Table 4. Per-Priority Class F1 Score Distribution for Augmentation Pipelines. Results suggest that, even though baseline outperforms the augmentation pipelines in priority scoring task, one of the main factors behind this is that the baseline shows high performance in classification on Low priority tweets, whereas the augmentation pipelines show a performance gain in the Critical, High, and Medium priority tweets.

Models	Critical	High	Medium	Low
Baseline	0.1205	0.2851	0.2974	**0.3792**
EDA	0.0904	**0.3265**	0.2937	0.1483
Syn-Aug	0.1052	0.2950	**0.3064**	0.2332
AugLy	**0.1432**	0.249	0.2985	0.2206

Table 4 showcases the F1 score for each priority class for the various augmentation pipelines. As evident from the tables, though baseline outperforms the rest of the pipelines in priority evaluation scores in Table 3, it outperforms the other pipelines in only Low priority class scores. AugLy outperforms the other augmentation pipelines in Critical priority class. EDA outperforms the other pipelines in High priority class score by a large margin. Though Synonym-Replacement pipeline outperforms the other pipelines in Medium priority class, the other pipelines are competitive.

Table 5. Per-Topic F1 Score Distribution for Augmentation Pipelines (Actionable Classes). Please note that these tables are transposed from Table 3 and 4 since the topics are more in number. Results show that EDA augmentation improves classification for actionable topics, however the mean performance for baseline and EDA augmentation pipelines remain same.

Topics	F1 Score			
	Baseline	EDA	Syn-Aug	AugLy
CallToAction-MovePeople	**0.58**	0.36	0.23	0.30
Report-EmergingThreats	0.20	**0.23**	**0.23**	0.22
Report-NewSubEvent	0.20	**0.21**	**0.21**	0.17
Report-ServiceAvailable	0.40	**0.46**	0.44	0.43
Request-GoodsServices	0.20	**0.33**	0.18	0.09
Request-SearchAndRescue	0.05	**0.06**	0.04	0.00
Mean	**0.27**	**0.27**	0.23	0.2

Tables 5 and 6 shows the results for individual information label for the three augmentation pipelines and also the baseline. EDA outperforms the other augmentation pipelines and the baseline in five out of six actionable classes and more than 60% of the non-actionable classes. A point to note is these tables also validate the point that baseline outperforms Synonym-Replacement and AugLy pipelines.

Table 6. Per-Topic F1 Score Distribution for Augmentation Pipelines (Non-Actionable Classes). Please note that these tables are transposed from Table 3 and 4 since the topics are more in number. Results show that the baseline outperforms the augmentation pipelines and shows the best results for non-actionable classes as well.

Topics	F1 Score			
	Baseline	EDA	Syn-Aug	AugLy
CallToAction-Donations	0.50	**0.53**	0.47	0.52
CallToAction-Volunteer	**0.31**	0.13	0.00	0.20
Other-Advice	**0.45**	0.42	0.40	0.42
Other-ContextualInformation	**0.11**	0.04	0.07	0.06
Other-Discussion	0.01	**0.06**	0.01	0.03
Other-Irrelevant	**0.67**	0.54	0.54	0.54
Other-Sentiment	0.35	**0.41**	0.39	0.37
Report-CleanUp	**0.57**	0.33	0.40	0.42
Report-Factoid	0.45	**0.52**	0.50	0.51
Report-FirstPartyObservation	0.12	**0.20**	0.19	0.17
Report-Hashtags	0.40	**0.52**	0.51	0.51
Report-Location	0.60	**0.63**	0.61	0.62
Report-MultimediaShare	0.32	0.36	0.37	**0.38**
Report-News	0.30	0.36	0.36	**0.37**
Report-Official	0.18	**0.21**	0.20	**0.21**
Report-OriginalEvent	**0.02**	0.01	0.01	0.01
Report-ThirdPartyObservation	**0.41**	0.12	0.17	0.15
Report-Weather	**0.66**	0.36	0.35	0.33
Request-InformationWanted	**0.63**	0.43	0.40	0.41
Mean	0.37	0.32	0.31	0.33

Model Selection

Table 7 compares the evaluation results for the three pre-trained model runs, namely, BERT, RoBERTa, and DeBERTa. As evident from the table, DeBERTa outperforms the rest in the information type classification task, with the exception of information accuracy, whereas RoBERTa outperforms the rest in priority scoring task.

Table 8 compares the F1 score on the basis of the four priority classes. As evident from the table, DeBERTa outperforms the other pipelines in Critical and Low priority classes, but showcases the worst performance in the High and Medium priority classes. RoBerta, though not performing well enough in Critical and Low priority classes, outperforms the rest of the model pipelines in the High and Medium priority classes.

Table 7. Evaluation Score Comparison Between Model Selection Pipelines. Results show that DeBERTa shows the best scores for information classification task, but shows performance loss in priority scoring task. RoBERTa shows the best scores in the priority scoring task and even showcases performance gain in information classification task.

Models	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline (BERT)	0.5043	0.1958	0.2872	0.8873	**0.2426**	0.2648	0.1978	0.229
RoBERTa	0.512	0.2475	0.3143	**0.8906**	0.2366	**0.2872**	**0.2043**	**0.2636**
DeBERTa	**0.5206**	**0.3436**	**0.3472**	0.8883	0.2355	0.2765	0.1247	0.2141

Table 8. Per-Priority Class F1 Score Distribution for Model Selection Pipelines. Results suggest that though DeBERTa shows best scores for Critical and Low priority tweets, it shows a performance loss in High and Medium priority tweets, as compared to the BERT baseline. However, RoBERTa shows a more consistent performance gain as compared to the BERT baseline, with showcasing the best results for High and Medium priority tweets, and just a slight loss in Critical priority scoring.

Models	Critical	High	Medium	Low
Baseline (BERT)	0.1205	0.2851	0.2974	0.3792
DeBERTa	**0.1636**	0.2757	0.2877	**0.6373**
RoBERTa	0.1109	**0.3099**	**0.3108**	0.5409

Table 9. Per-Topic F1 Score Distribution for Model Selection Pipelines (Actionable Classes). Please note that these tables are transposed from Table 7 and 8 since the topics are more in number. Results show that DeBERTa outperforms the other two pipelines in F1 score for actionable classes.

Topics	F1 Score		
	Baseline (BERT)	RoBERTa	DeBERTa
CallToAction-MovePeople	0.32	0.46	**0.52**
Report-EmergingThreats	**0.23**	0.20	0.22
Report-NewSubEvent	0.17	0.14	**0.19**
Report-ServiceAvailable	0.40	0.49	**0.50**
Request-GoodsServices	0.01	0.08	**0.50**
Request-SearchAndRescue	0.04	**0.13**	**0.13**
Mean	0.20	0.25	**0.34**

Tables 9 and 10 compares the F1 scores from individual information type labels. For F1 scores for Actionable classes, as shown in Table 9, DeBERTa outperforms the rest of the pipelines in five of the six information classes. For the F1 score for non-actionable classes as well, as shown in Table 10, DeBERTa outperforms the rest of the pipelines in almost 70% of the information labels.

Table 10. Per-Topic F1 Score Distribution for Model Selection Pipelines (Non-Actionable Classes). Please note that these tables are transposed from Table 7 and 8 since the topics are more in number. Results show that DeBERTa outperforms the other two pipelines in F1 score for non-actionable classes as well.

Topics	F1 Score		
	Baseline (BERT)	RoBERTa	DeBERTa
CallToAction-Donations	**0.53**	0.43	0.49
CallToAction-Volunteer	0.13	0.14	**0.29**
Other-Advice	0.41	**0.43**	0.41
Other-ContextualInformation	**0.07**	0.06	**0.07**
Other-Discussion	0.02	**0.05**	**0.05**
Other-Irrelevant	0.54	**0.60**	**0.60**
Other-Sentiment	0.40	**0.41**	**0.41**
Report-CleanUp	0.26	**0.51**	0.49
Report-Factoid	0.51	**0.53**	0.48
Report-FirstPartyObservation	0.17	**0.21**	0.20
Report-Hashtags	0.50	**0.54**	**0.54**
Report-Location	**0.62**	0.60	0.56
Report-MultimediaShare	0.37	0.39	**0.40**
Report-News	**0.36**	**0.36**	**0.36**
Report-Official	**0.20**	0.13	**0.20**
Report-OriginalEvent	0.00	**0.01**	**0.01**
Report-ThirdPartyObservation	**0.17**	0.14	0.14
Report-Weather	**0.36**	0.33	0.29
Request-InformationWanted	0.38	0.51	**0.63**
Mean	0.32	0.34	**0.35**

Learning Methodology

This section shows the results obtained for the experiment between a single multi-task pipeline, and two separate task-specific pipelines, one for each information label classification task and as well as one for priority scoring task. In the Tables 11, 13, and 14, we discuss these two pipelines, with “Separate Task" representing the pipeline using two separate task-specific models, and “Multi-Task” representing the multi-task learning pipeline.

As evident from Table 11, even though multi-task learning is competitive in information type classification and showcases performance gain in nDCG and Priority F1 for all metrics, we see a general loss in performance when using multi-task learning method.

Table 11. Evaluation Score Comparison Between Learning Methodology Pipelines. Results show that the baseline, which uses separate-task learning, outperforms multi-task learning in most of the evaluation scores except nDCG and Priority F1 for All classes..

Leanring Stratergies	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline (Seperate Task)	0.5043	**0.1958**	**0.2872**	**0.8873**	**0.2426**	0.2648	**0.1978**	**0.229**
Multi-Task	**0.5426**	0.1947	0.2769	0.8857	0.2169	**0.2787**	0.177	0.2255

Table 12 showcases the F1 score split for each priority label for the two pipelines using different learning methods. Even though separate-task learning outperforms multi-task learning in all evaluation priority scores, as evident from Table 11, the separate-task learning is only better in High and Medium priority labels, and multi-task learning outperforms in Critical and Low priority labels.

Table 12. Per-Priority Class F1 Score Distribution for the Two Learning Methodology Pipelines. Results suggest that one of the main reasons for the better performance shown by separate-task learning can be the higher performance in scoring High and Medium priority tweets, whereas multi-task learning pipeline outperforms separate-task learning in Critical and Low priority tweets.

Learning Statergies	Critical	High	Medium	Low
Baseline (Seperate Task)	0.1205	**0.2851**	**0.2974**	0.3792
Multi-Task	**0.1955**	0.255	0.2829	**0.4838**

Table 13. Per-Topic F1 Score Distribution for Learning Methodology Pipelines (Actionable Classes). Please note that these tables are transposed from Table 11 and 12 since the topics are more in number. Results suggest that even though multi-task learning shows best performance in 50% of the actionable classes, on average, separate-task learning shows slightly higher performance than multi-task learning on actionable classes.

Topics	F1 Score	
	Baseline	Multi-Task
CallToAction-MovePeople	**0.32**	0.29
Report-EmergingThreats	**0.23**	0.2
Report-NewSubEvent	**0.17**	0.16
Report-ServiceAvailable	0.4	**0.45**
Request-GoodsServices	0.01	**0.02**
Request-SearchAndRescue	**0.04**	**0.04**
Mean	**0.2**	0.19

Tables 13 and 14 shows the evaluation results for individual information labels for the two mentioned pipelines. Even though multi-task learning is competitive in actionable classes scores, as evident from Table 13, single-task learning showcases the best results in more than 70% of the non-actionable information classes, as evident from Table 14.

Table 14. Per-Topic F1 Score Distribution for Learning Methodology Pipelines (Non-Actionable Classes). Please note that these tables are transposed from Table 11 and 12 since the topics are more in number. Results show that, unlike for actionable classes, separate-task learning clearly outperforms multi-task learning in majority of non-actionable classes.

Topics	F1 Score	
	Baseline	Multi-Task
CallToAction-Donations	**0.53**	0.47
CallToAction-Volunteer	**0.13**	0.04
Other-Advice	0.41	**0.43**
Other-ContextualInformation	**0.07**	**0.05**
Other-Discussion	0.02	**0.03**
Other-Irrelevant	**0.54**	0.53
Other-Sentiment	**0.40**	**0.40**
Report-CleanUp	**0.26**	0.22
Report-Factoid	**0.51**	0.50
Report-FirstPartyObservation	**0.17**	**0.17**
Report-Hashtags	**0.50**	0.49
Report-Location	**0.62**	0.61
Report-MultimediaShare	0.37	**0.38**
Report-News	**0.36**	**0.36**
Report-Official	**0.20**	0.19
Report-OriginalEvent	0.00	**0.01**
Report-ThirdPartyObservation	0.17	**0.18**
Report-Weather	**0.36**	0.34
Request-InformationWanted	**0.38**	0.33
Mean	**0.32**	0.3

Section Prioritization

This section compares the results from the other sections and the insights gained from those sections and showcases some pipelines which prioritize one section over the other with the aim to get better performance gain. As discussed in the Introductions section, we use various abbreviations for the various pipelines discussed in this section.

Table 15. Evaluation Scores Comparison for Section Prioritization Pipeline. Results show that our earlier experiment with the BERT model over non-augmented imbalanced data for multi-task learning showcased performance loss, more sophisticated and denser models like DeBERTa show higher performance with multi-task learning than separate-task learning, as evident from scores for M-DeEDA and S-DeEDA, with M-DeEDA even showcasing best results in nDCG and Information Classification F1 score for actionable classes. S-RoEDA also showcases the best results for both the Priority F1 scores.

Model Description	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Baseline (BERT)	0.5043	0.1958	0.2872	0.8873	0.2426	0.2648	0.1978	0.229
Best of RQ1	0.5043	0.2728	0.312	0.8873	0.2586	0.2648	0.1978	0.229
Best of RQ2	0.512	0.3436	0.3472	0.8906	0.2366	0.2872	0.2043	0.2636
Best of RQ3	0.5426	0.1958	0.2872	0.8873	0.2426	0.2787	0.1978	0.229
S-RoEDA	0.5252	0.288	0.3215	0.891	**0.264**	**0.2976**	0.2117	0.282
M-RoEDA	0.5433	0.2999	0.3182	0.8925	0.25	0.2763	0.2084	**0.2995**
S-DeEDA	0.5133	0.3288	0.3437	0.8877	0.2337	0.2893	0.1319	0.2189
M-De	0.5424	0.3238	0.3382	0.8869	0.2432	0.2829	0.1908	0.2918
M-DeEDA	**0.5579**	**0.3449**	0.3443	0.8854	0.2398	0.2745	0.205	0.273
Best Run Results	**0.5579**	0.3311	**0.3555**	**0.8931**	0.2612	0.297	**0.2603**	0.2982

Table 17. Per-Priority Class F1 Score Distribution for the Various Section Prioritization Pipelines. Results suggest that RoBERTa in general showcases best performance in scoring Critical, High, and Medium priority tweets, however S-DeEDA showcases the best performance in scoring Low priority tweets.

Model Description	Critical	High	Medium	Low
Baseline (BERT)	0.1205	0.2851	0.2974	0.3792
S-RoEDA	0.1878	**0.33**	**0.3198**	0.5017
M-RoEDA	**0.2201**	0.267	0.2769	0.4719
M-De	0.2037	0.2757	0.2882	0.4915
S-DeEDA	0.1803	0.2842	0.2586	**0.6516**
M-DeEDA	0.2164	0.2756	0.2971	0.4744

Table 16. Percent Improvement In Model Selection Pipelines in Comparison to Baseline. Results shows that using multi-task learning with DeBERTa gives improved performance in general, with M-DeEDA showcasing maximum performance gain in nDCG and Info Type Actionable F1 Scores. Using RoBERTa in general showcases a consisted performance gain, with maximum performance gain in almost all evaluation scores for priority scoring task.

Model Description	nDCG@100	Info Type F1		Info Accuracy	Priority F1		Priority R	
		Actionable	All		Actionable	All	Actionable	All
Best of RQ1	-	39.33	8.64	-	6.60	-	-	-
Best of RQ2	3.23	75.49	**20.89**	0.37	-2.47	8.46	3.29	15.11
Best of RQ3	7.59	-	-	-	-	5.25	0.00	-
S-RoEDA	4.14	47.09	11.94	0.42	**8.82**	**12.39**	**7.03**	23.14
M-RoEDA	7.73	53.17	10.79	**0.59**	3.05	4.34	5.36	**30.79**
S-DeEDA	1.78	67.93	19.67	0.05	-3.67	9.25	-33.32	-4.41
M-De	7.56	65.37	17.76	-0.05	0.25	6.84	-3.54	27.42
M-DeEDA	**10.63**	**76.15**	19.88	-0.21	-1.15	3.66	3.64	19.21

Tables 15 and 16 showcase the results for the various pipelines mentioned. As evident from the tables, DeBERTa with EDA augmentation and multi-task learning showcase great performance gain in information classification task, with even outperforming the best run submitted to TREC-IS by a wide margin in the Information Type F1 score for actionable classes. However, RoBERTa in general shows better performance gain in priority scoring task, with RoBERTa with EDA augmentation and separate-task learning outperforming the best run submitted to TREC-IS in Priority F1 scores, for both actionable as well as all classes.

Table 16 also showcases that using DeBERTa sometimes gives loss in performance for priority scoring task, as evident for DeBERTa + Separate + EDA pipeline. However, switching learning method from separate-task learning to multi-task learning does seem to improve the performance, except for Priority F1 score for actionable classes.

Table 17 showcases the F1 score for individual priority classes for the different pipelines used in section selection experiments. Results from Table 17 support the results from Tables 15 and 16, providing further evidence that RoBERTa shows better performance in priority scoring task as compared to DeBERTa model.

Table 18. Per-Topic F1 Score Distribution for Section Prioritization Pipelines (Actionable Classes). Please note that these tables are transposed from Table 15 and 17 since the topics are more in number. Results show that even though no pipeline has a clear majority in best scores for actionable class classification, DeBERTa in general shows best performance for information classification task.

Topic	F1 Score					
	Baseline	S-RoEDA	M-RoEDA	S-DeEDA	M-De	M-DeEDA
CallToAction-MovePeople	0.32	0.43	0.37	0.42	0.46	**0.52**
Report-EmergingThreats	0.23	0.20	0.23	0.22	**0.25**	0.23
Report-NewSubEvent	0.17	0.17	0.19	0.16	0.21	**0.22**
Report-ServiceAvailable	0.40	0.51	**0.52**	**0.52**	0.50	0.50
Request-GoodsServices	0.01	0.36	0.34	**0.50**	0.39	0.44
Request-SearchAndRescue	0.04	0.05	0.14	0.14	0.14	**0.15**
Mean	0.20	0.29	0.30	0.33	0.32	**0.34**

Table 19. Per-Topic F1 Score Distribution for Section Prioritization Pipelines (Non-Actionable Classes). Please note that these tables are transposed from Table 15 and 17 since the topics are more in number. Results, similar to results for actionable classes, though no pipeline has a clear majority in best scores for non-actionable class classification, DeBERTa in general shows best performance for information classification task.

Topic	F1 Score					
	Baseline	S-RoEDA	M-RoEDA	S-DeEDA	M-De	M-DeEDA
CallToAction-Donations	**0.53**	0.44	0.41	0.46	0.50	0.46
CallToAction-Volunteer	0.13	0.07	0.03	**0.30**	0.26	0.29
Other-Advice	0.41	0.40	**0.43**	0.41	0.42	0.40
Other-ContextualInformation	0.07	0.05	0.07	0.06	**0.08**	0.05
Other-Discussion	0.02	0.04	0.02	**0.05**	0.03	0.04
Other-Irrelevant	0.54	0.59	**0.60**	0.57	0.56	0.55
Other-Sentiment	0.40	0.42	0.43	**0.44**	0.42	0.43
Report-CleanUp	0.26	**0.52**	0.45	0.49	0.49	0.49
Report-Factoid	**0.51**	0.47	0.48	**0.51**	0.50	**0.51**
Report-FirstPartyObservation	0.17	**0.23**	**0.23**	0.19	0.16	0.17
Report-Hashtags	0.50	**0.55**	**0.55**	0.56	0.54	**0.55**
Report-Location	**0.62**	0.59	0.58	0.59	0.57	0.60
Report-MultimediaShare	0.37	0.37	0.37	**0.41**	**0.41**	0.40
Report-News	**0.36**	0.35	0.33	0.35	**0.36**	**0.36**
Report-Official	**0.20**	0.16	0.17	**0.20**	0.18	**0.20**
Report-OriginalEvent	0.00	**0.02**	0.01	0.01	0.01	0.00
Report-ThirdPartyObservation	0.17	0.12	0.10	0.14	**0.18**	0.16
Report-Weather	**0.36**	0.35	0.35	0.31	0.27	0.30
Request-InformationWanted	0.38	0.56	0.56	0.55	0.56	**0.59**
Mean	0.32	0.33	0.32	**0.35**	0.34	0.34

Tables 18 and 19 show the F1 score split for each topic for the various section selection pipelines. As evident from the table even though DeBERTa with EDA augmentation and multi-task learning(M-DeEDA) does seem to give the best results in 50% of the actionable classes, it does not perform as well in non-actionable classes, as observed in Table 19. DeBERTa in general gives better result as compared to RoBERTa.

DISCUSSION

This section aims to discuss and draw inferences from the results presented in the previous section.

Augmentation Comparison

For this experiment, we compared three different augmentation methods, namely, EDA, Synonym-Replacement, and AugLy. As discussed in the results, all three augmentation method show a general improvement in the information classification task, with EDA outperforming the other pipelines. However, there is a performance loss in all three augmentation pipelines for the priority scoring task. However, upon analysis of per-priority F1 scores, it is evident that the baseline outperforms in only the Low priority classes, and augmentation with any of the methods showcases improvement in the Critical and High priority classes. The analysis of per-topic F1 scores presents us the insight that even though there is no clear winner among the augmentation methods, EDA showcases highest F1 scores in majority of information classes, for both actionable as well as non-actionable classes.

Model Selection

For this experiment, we compared three different language models, namely, BERT, RoBERTa, and DeBERTa. As evident from the results, DeBERTa showcases a massive performance gain in the information classification task as compared to the baseline (BERT) and the RoBERTa pipelines, but shows loss in performance in the priority scoring task as compared to the baseline. However, the RoBERTa pipeline outperforms the other two pipelines in the priority scoring task but also shows performance gain in information classification task as compared to the baseline. Analysis of per-priority class F1 score showcases DeBERTa outperforming in Critical and Low priority classes, with a performance loss in High and Medium priority classes, as compared to baseline. Thus, the performance gain showcased by RoBERTa in priority scoring task can be attributed to the high performance in High and Medium priority classes and competitive F1 scores in the other two priority classes. The analysis of per-topic F1 score supports the performance gain in information classification task by DeBERTa, with the DeBERTa showcasing the highest F1 scores in majority of actionable as well as non-actionable information classes. Thus, from these results we can draw the inference that, though state-of-the-art off-the-shelf models like DeBERTa do show performance gain in information classification task as compared to the baseline BERT model, but models like RoBERTa, which, though is more denser and more complex than the BERT baseline model, but not as complex as the state-of-the-art DeBERTa model, shows a more consistent performance gain across both of the tasks (information classification and priority scoring).

Learning Methodology

For this experiment, we compared two different learning methodologies, separate-task learning and multi-task learning. We define separate-task learning as training two different models, one for information classification only and one for priority scoring only. Multi-task learning is described as training one model which uses the loss functions from both information classification task and priority scoring task. As evident from the results, separate-task learning outperforms multi-task learning in general, except for nDCG score and the priority F1 score for all classes. Further analysis of per-priority class F1 scores showcases an improvement in Critical and Low priority classes by multi-task learning pipeline, with a performance loss in High and Medium priority classes. Analysis of per-topic F1 scores also gives evidence of no significant improvement by multi-task learning pipeline. From these results, we can infer that separate-task learning is a much more consistent as compared to multi-task learning. However, for cases where there is a higher need of classifying Critical priority tweets with higher accuracy, multi-task learning might be taken into consideration. Also, using a basic language model such as BERT might be reducing the amount of information gained from introducing loss from both the tasks. To further analyze this we compare different models in the Section Prioritization experiments.

Section Prioritization

In this work, we compare various pipelines using different methods for the three major sections of a pipeline, namely, data augmentation, language model selection, and learning methodology. From the individual pipeline results discussed above, we can conclude that using EDA for data augmentation with DeBERTa as the language model and separate-task learning as the learning methodology should give us the best results for information classification task and using EDA for data augmentation with RoBERTa as the language model and separate-task learning as the learning methodology should give us the best results for priority scoring task. However, the results for Section Analysis pipelines, contrary to expectations, showcase that DeBERTa model with EDA augmentation and multi-task learning (M-DeEDA) outperforms DeBERTa model with EDA augmentation and separate-task learning (S-DeEDA) in information classification task as well as majority of priority scoring task, and M-DeEDA also outperformed the best run submitted to TREC-IS in information type F1 score for actionable classes. However, RoBERTa with EDA augmentation and separate-task learning (S-RoEDA) does give us the best evaluation scores in almost all scores for priority scoring task except for Priority R for all classes, with S-RoEDA even outperforming the best run results in both the Priority F1 scores.

Further analysis of percentage gain of each pipeline as compared to the baseline showcases that upon using a denser model like DeBERTa with multi-task learning (M-De) we see some performance gain in the priority scores, which was not the case for BERT model. Training DeBERTa on an EDA augmented dataset with multi-task learning also improves the performance in the evaluation score for priority scoring tasks. Thus we can infer that, multi-task learning does show performance gain, but only in denser models like DeBERTa. Analysis of per-priority class F1 scores confirms our expectations with RoBERTa, in general, showcasing the best results for Critical, High and Medium priority classes, with multi-task learning showcasing improved performance for Critical priority tweets and separate-task learning showcasing higher performance for High and Medium priority tweets. However, upon analysis of per-topic F1 scores, we see no best pipeline which outperforms the rest. DeBERTa with EDA augmentation and multi-task learning (M-DeEDA) does give the best results in half of the actionable classes, however, DeBERTa in general does showcase maximum best results among its three pipeline, M-De, S-DeEDA, and M-DeEDA.

Thus, we can infer that even though multi-task learning does not perform well with models like BERT and RoBERTa, state-of-the-art models like DeBERTa show certain performance gain when employing multi-task learning over separate-task learning. High end models like DeBERTa in general are better for information classification task, whereas model like RoBERTa are better for priority scoring task.

CONCLUSION

In this work, we aimed to compare and analyze different methods for the three major sections of a pipeline, namely, augmentation method, language model selection, and learning methodology, as presented in Figure 1. We outlined a baseline model architecture which used the raw textual data without any augmentation, BERT as its language model and separate-task learning, as presented in Figure 5. In this work, we aimed to answer four core research questions.

RQ1: *Across several augmentation techniques, what is the expected performance increase, on average and for each method, when applied to crisis-informatics data?*

During our initial experimentation with BERT as the core language model, we show that though augmenting data with EDA does improve results for information classification task, there is a general performance loss in priority scoring task when using augmented data as compared to using raw, non-augmented data. However, on a deeper analysis of F1 scores for individual priority classes, we infer that the reason for higher performance showcased by baseline using non-augmented dataset is due to a higher performance in scoring Low priority tweets, which are higher in number. As evident from Table 4, augmentation improves the performance for Critical, High and Medium priority tweets. Thus, we can say that augmentation does showcase performance gain.

RQ2: *By how much might one expect performance to increase by using increasingly sophisticated*
off-the-shelf,
pre-trained models?

We experimented with two language models, one state-of-the-art, off-the-shelf language model DeBERTa, and the other an improved version of our base model RoBERTa, in comparison with the base BERT model. Our experiments showcased that though state-of-the-art models like DeBERTa do improve performance for information classification task, there is a performance loss in priority scoring task. Models like RoBERTa are much more consistent in performance gain, with RoBERTa outperforming DeBERTa as well as our baseline(BERT) in priority scoring task and showcased performance gain in information classification task as well. Thus, we can conclude that though state-of-the-art models like DeBERTa do show performance gain, one might consider the task which needs to be prioritized, information classification, for which the results recommend state-of-the-art models like DeBERTa, or priority scoring, for which the results recommend a simpler yet efficient model like RoBERTa.

RQ3: *How much performance increase is observed upon using a single multi-task learning pipeline, as compared to different task-specific pipelines?*

We experiment with two learning methodologies, one using separate task-specific models, and one using a single multi-task learning pipeline with shared weights. Our initial experiments with BERT showcase that separate task-specific models show higher performance as compared to multi-task learning, as presented in Table 11. However, using a more sophisticated model like DeBERTa showcases improved performance for multi-task learning pipeline, as presented in Table 15. Thus, we can conclude that multi-task learning showcases performance gain for sophisticated models like DeBERTa and thus the core language model selected should be taken into consideration when selecting a learning methodology.

RQ4: *Across data augmentation, language model selection, and training methodology, which choices should be prioritized for optimization for maximum performance gain?*

We compared various pipelines with a varied use of EDA augmentation, DeBERTa and RoBERTa language models and separate- as well as multi-task learning methodologies against the best results from the experiments for RQ1, RQ2, and RQ3, and also from the best results from run submissions to TREC-IS 2021-A. Through our experiments, as presented in Tables 15 and 16, we can conclude that when prioritizing a section, the task should be taken into consideration. For information classification task, augmentation and model selection should be prioritized over learning methodology while for priority scoring task learning methodology and model selection should take prioritization. Our results suggest that sophisticated models like DeBERTa with EDA augmentation and multi-task learning perform better on information classification task, while simpler models like RoBERTa with EDA augmentation and separate-task learning perform better for priority scoring task.

REFERENCES

Bishop, C. M. (1995). “Training with noise is equivalent to Tikhonov regularization”. In: *Neural computation* 7.1, pp. 108–116.

Buntain, C., McCreadie, R., and Soboroff, I. (2020). “Incident Streams 2020: TRECIS in the Time of COVID-19”. In: *Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020), Gaithersburg, Maryland.*

Caragea, C., Silvescu, A., and Tapia, A. H. (2016). "Identifying informative messages in disaster events using convolutional neural networks". In: *International conference on information systems for crisis response and management*, pp. 137–147.

Chowdhury, J. R., Caragea, C., and Caragea, D. (2020). "Cross-lingual disaster-related multi-label tweet classification with manifold mixup". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 292–298.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). "Electra: Pre-training text encoders as discriminators rather than generators". In: *arXiv preprint arXiv:2003.10555.*

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805.*

He, P., Liu, X., Gao, J., and Chen, W. (2020). "Deberta: Decoding-enhanced bert with disentangled attention". In: *arXiv preprint arXiv:2006.03654.*

Kobayashi, S. (2018). "Contextual augmentation: Data augmentation by words with paradigmatic relations". In: *arXiv preprint arXiv:1805.06201.*

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). "Albert: A lite bert for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942.*

Liu, J., Singhal, T., Blessing, L. T., Wood, K. L., and Lim, K. H. (2021). "Crisisbert: a robust transformer for crisis classification and contextual crisis embedding". In: *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, pp. 133–141.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692.*

Neppalli, V. K., Caragea, C., and Caragea, D. (2018). "Deep neural networks versus naive bayes classifiers for identifying informative tweets during disasters". In: *Proceedings of the 15th Annual Conference for Information Systems for Crisis Response and Management (ISCRAM).*

Nguyen, D. T., Mannai, K. A. A., Joty, S., Sajjad, H., Imran, M., and Mitra, P. (2016). "Rapid classification of crisis-related data on social networks using convolutional neural networks". In: *arXiv preprint arXiv:1608.03902.*

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108.*

Sun, C., Gong, Y., Wu, Y., Gong, M., Jiang, D., Lan, M., Sun, S., and Duan, N. (2019). "Joint type inference on entities and relations via graph convolutional networks". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1361–1370.

Wang, C., Nulty, P., and Lillis, D. (2021). "Transformer-based Multi-task Learning for Disaster Tweet Categorisation". In: *Proceedings of the International ISCRAM Conference* 2021-May.May.

Wei, J. and Zou, K. (2019). "Eda: Easy data augmentation techniques for boosting performance on text classification tasks". In: *arXiv preprint arXiv:1901.11196.*

Xue, K., Zhou, Y., Ma, Z., Ruan, T., Zhang, H., and He, P. (2019). "Fine-tuning BERT for joint entity and relation extraction in Chinese medical text". In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).* IEEE, pp. 892–897.

Zhang, X., Zhao, J., and LeCun, Y. (2015). "Character-Level Convolutional Networks for Text Classification". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1.* NIPS'15. Montreal, Canada: MIT Press, pp. 649–657.

Zhang, Y. and Yang, Q. (2017). "A survey on multi-task learning". In: *arXiv preprint arXiv:1707.08114.*

Zheng, S., Hao, Y., Lu, D., Bao, H., Xu, J., Hao, H., and Xu, B. (2017). "Joint entity and relation extraction based on a hybrid neural network". In: *Neurocomputing* 257, pp. 59–66.

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., and Xu, B. (2017). "Joint extraction of entities and relations based on a novel tagging scheme". In: *arXiv preprint arXiv:1706.05075.*